Commodore

The Transactor

comments and bulletins concerning your COMMODORE PET™

BULLETIN #7
December 31, 1978

PET™ is a registered Trademark of Commodore Inc.

Yes, it's finally here! Included in this issue of the "Transactor" are chapters 1 through 7 of the new PET Users Manual. Chapters 8 through 9 and the appendices will be made available in the next bulletin, January 1979. This release will cover November as well as December.

Karl J.

STANDARD SYMBOLS

I have assembled a small table of symbols that are not hard to obtain from a typewriter (if you are using one) and are quite distinguishable if you write out your programs in capital block letters. It would be appreciated if you use these when submitting software for publication; especially programs containing cursor control.

SYMBOLS

h - cursor home	¢ - carriage return
c - clear screen	ø - space (blank)
cr- cursor right	! - RVS on
cl- cursor left	/ - RVS off
cu- cursor up	@ - Shift on
cd- cursor down	
d - delete	
i - insert	

To represent any graphic characters clearly, the character below should be followed by the '@'. For example:

▧ = M@ (M w/Shift on)

Following this standard should make programs fairly easy to read however any suggestions are quite welcome.

Karl J.

Commodore would like to take this opportunity to announce the appointment of Mr. Karl J. Hildon as our new Technical Advisor for the Computer Division.

Mr. Hildon has a Diploma in electronics technology and has taken some courses in computer programming. He will be in charge of answering any software and/or hardware inquiries pertaining to the PET computer and upcoming peripherals. If you have any questions, please do not hesitate to call Mr. Hildon at Commodore or write:

Karl J. Hildon
Commodore Business Machines Ltd
Computer Division
3370 Pharmacy Avenue
Agincourt, Ont. M1W 2K4
Tel: 499-4292.

Karl is also actively interested in the interfacing aspect of the PET computer and would like to learn more as time progresses. He has therefore written the following letter which will be included with any hardware information he sends out and has asked that it be inserted here.

"Dear Interfacers:

Your requested User Port and/or IEEE-488 Bus information has arrived! Should you find the enclosed data insufficient, please call or write Commodore. However, if it does assist you to a solution, please write in with any interesting discoveries or ideas. They would be much appreciated and may find their way into a "Transactor" bulletin!

Yours sincerely,

Karl Hildon
Technical Advisor
Computer Division

P.S. All published articles will get author recognition."

Mr. Hildon will also be, for the most part, our "Transactor" bulletin editor. Therefore, letters, articles, software, circuitry, discoveries or anything at all that you feel you would like to see published in our "Transactor" should be addressed to:

Karl J.
Commodore Business Machines Ltd.
"Transactor" bulletin
3370 Pharmacy Avenue
Agincourt, Ont. M1W 2K4

Brief replies will be included in the "Transactor" following any letters submitted for the "Transactor" and any articles will receive full recognition of the author provided they haven't been published previously or elsewhere.

In the meantime, your enthusiasm and interest in the PET computer is most refreshing. Keep it coming!

NEWSLETTER ADDRESSES

1. The "Tranasctor" (subs. $10.00 Cdn.)
   3370 Pharmacy Avenue
   Agincourt, Ont.
   M1W 2K4

2. PET User Notes (subs. $6.00 U.S.)
   P.O. Box 371
   Montgomeryville, PA 18936

3. THE PET PAPER (subs. $15.00 U.S.)
   P.O. Box 43
   Audubon, PA 19407

4. Cursor (subs. $24.00 U.S.)
   P.O. Box 550
   Goleta, Calif. 93017

5. The PET Gazette (Free w/lg. S.A.S.E. U.S. postage)
   c/o Len Lindsay
   1929 Northport Drive
   Room 6
   Madison, Wisconsin 53704

TAPE LIBRARY

Club 2001, the Toronto PET Users Group, now has an active tape library. The deal is $3.00 deposit which is refunded upon return. The only drawback is that access to the tapes is limited to club meetings and evenings at Lyman Duggan's, the Club co-ordinator. Lyman and I have made arrangements to make copies of the entire library, which incidentally stands at 70 and growing, and make them available through the day. Anyone who wishes to obtain a library program is invited to come to the Commodore office in Agincourt and make a copy. Please bring a blank tape with and any other programs you might like to add to the library.

Karl J.

Peripherals

The Disc Drive and Printers are coming. I know you've all heard that before but we'll be announcing a date January 6 as to availability and so forth. Those of you who purchase and wish to convert to Floppy Disc storage can get the DOS ROMs free of charge once available. The Printers are directly interfacable to the PET and will have 'intelligence' such that they will accept formatting commands for business applications, etc. More detailed descriptions and specifications will be available from Commodore as of Jan. 6 and will be published in the next Transactor.

Karl J.

Le PET

An instruction course on the PET is being offered at Irisco du Quebec and will be taught in French. Cost of the programme will be $75.00 which will also cover manuals and notes. The course is 15 hours long plus additional practice time and will begin the first week of January, 1979. Anyone interested should contact:

Mr. Ricardo Talbot
Irisco du Quebec
337 Boul. Charest Est.,
Quebec City, Que. G1K 3J2
Tel: 418-647-4422

Also, Mr. Talbot has a tape available containing 11 games, all of which are in French.

Un cours D'instruction en Francais sera a votre disposition chez Irisco du Quebec. Le prix du cours est $75.00 qui comprent des notes et des manuels. Le cours va commencer dans la premiere semaine du mois de janvier 1979 et va couvrir 15 heures de leçons, en plus le temps pour la practique. Si vous etes interesses contactez svp:

M. Ricardo Talbot
Irisco du Quebec
337 Boul. Charest Est.,
Quebec City, Que. G1k 3J2
Tel: 418-647-4422

M. Talbot aura aussi a votre disposition les cassettes contenant 11 jeux tous en Francais.

FLASH

Richvale Telecommunications,10610 Bayview Ave; Richmond Hill,Ontario L4C 3N8 884-4165 are pleased to announce two new interfaces for the Pet 2001-8.
"Hampet": Amorse and RTTY interface.
"Modpet": For sending computer information over the phone.
The "Hampet" is a morse and RTTY(Baudot) interface.You simply plug the "Hampet" into the Pet,and your ready to display, transmit and receive morse and RTTY at rates upto 100WPM.
The interface will also send random words and characters, displayed and with audio from a built in side tone oscillator and speaker.In addition the "Hampet" can re-transmit 400 characters from memory for "brag" transmissions.
The "Hampet" is sold wired and tested,and comes in an attractive,blue and sand,sloping front,aluminum case,with on-off,volume control,frequency control,PLL control,tune LED,on LED,headphone muting jack and four RCA phone jacks.
The "Hampet" is capable of perfect copy with an "S4" signal.

The "Modpet" plugs into the Pet and is interfaced to the red and green phone wires,the audio level is adjusted with a VU meter and the user can trade tapes and play interactive games over the telephone.The "Modpet" is not directly audio coupled into the line resulting in superb send and receive.
It is suggested that potential buyers first check with their local Bell office,for in spite of recent court rulings,Bell may insist on installing a coupler.The "Modpet" is not suitable for long distance due to distortion and the narrower band width created by "Ma Bell".
The "Hampet" retails for 189 dollars with immediate delivery.
The "Modpet" retails for 79 dollars with three week delivery.

Dealer enquiries are invited.
Ben Arnold VE3EYB Richvale Telecommunications.

Jim Butterfield, well known PET enthusiast, has sent to Commodore a couple of interesting items which have also been passed on to PET User Notes. Thanks Jim...and thanks for recognizing Brad as a source--maybe he'll send in something else as well. -Karl J.

Summary of Cassette data file "patches"

The following information has been passed around users, and is now "official" with the issuance of a Commodore bulletin. It seems worth while to summarize briefly:

I. Opening a file for writing: an omission in current ROM programs makes it highly desirable to precede all OPEN statements with a couple of POKEs:

before OPEN x,1,.. for writing: POKE 243,122 : POKE 244,2
before OPEN x,2,.. for writing: POKE 243,58 : POKE 244,3

2. When writing tapes, it is useful to increase the spacing between tape blocks; otherwise you might miss a block during subsequent reading. There are several approaches to this; my technique is to call the following subroutine immediately after each PRINT#:

Cassette #1

```
950 IF Z9<=PEEK(625)GOTO 990
960 POKE 59411,53
970 FOR Z9=1 TO 60:NEXT Z9
980 POKE 59411,61
990 Z9=PEEK(625):RETURN
```

Cassette #2

```
950 IF Z9<=PEEK(626)GOTO 990
960 POKE 59456,207
970 FOR Z9=1 TO 60: NEXT Z9
980 POKE 59456,223
990 Z9=PEEK(626):RETURN
```

3. Even with the above coding, it seems wise to guard against a potential "dropped block". Think in terms of writing a "number of items" total on tape so that when reading, you can check that nothing has been lost.

4. Don't PRINT# a line of over 80 characters unless you're prepared to do some careful work with GET# statements when you read it back in. In general, avoid "print punctuation" when writing (PRINT#1, A;B ... PRINT#2, X$,Y$); each data element can be written as a separate "line". Watch for long strings.

5. Either: check the value of ST after <u>every</u> read, or use your own checking routines on your data. ST can be useful, but doesn't guarantee your data is 100% good. IF ST=0 .. no errors are seen; IF ST>63 .. you have come to the end of file; if anything else, an error has been detected.

6. Always CLOSE your cassette files after you're finished with them. When writing, your data is accumulated into a buffer .. if you don't CLOSE, it may not go onto tape.

Most Basic errors abort the cassette file without CLOSEing it; if this happens while you have a cassette tape open for writing, better start over .. your tape will likely have data missing.

UNLIST Revisited — Jim Butterfield, Toronto

Larry Tessler's UNLIST appeared in User Notes without much fanfare. If you can read between the lines, however (and cope with the typos) it's quite a blockbuster of a program.

In general, it allows a program to be handled as data ... using the UNLIST key, you can re-process programs as if they were data files, and create such things as program-writing programs, language translators, and many other startling things.

Perhaps the most immediate use of UNLIST for the casual computerist is to merge two programs together. One program could contain subroutines; and these could be merged with other programs to save a lot of typing. It's especially useful to be able to merge a single set of DATA statements into several programs, each of which is set up to process the DATA in different ways.

Brad Templeton (Toronto) has passed me an even more concise way of doing the same thing. I'll go through the whole operation, step by step.

First, prepare the program you will want to merge in the following manner. Load the program. Put a blank tape into the cassette and rewind. Enter OPEN 1,1,1 : CMD 1 : LIST. Be sure to put this on a single line, using colons as indicated. Press RECORD and PLAY as instructed. The tape will move. When it stops, type ?"POKE611,0": PRINT#1:CLOSE1. Your tape is now ready, and PET should be back to normal operation. You may file this tape and use it at any future time.

Now for the merge. When you have your second program loaded into the PET, mount the tape you have previously written. Type OPEN1, press PLAY as requested, and wait for the tape to stop.

Here comes the tricky bit. Clear the screen, give 4 cursor down's, and type the following line, but DO NOT HIT RETURN:

POKE611,1:POKE525,1:POKE527,13:?"h" (h is Cursor Home, displaying reverse S)

Don't hit return. Instead, press cursor home and 6 cursor downs ... then type the identical line. This time, hit RETURN at the end of the line and listen to the tape move.

Eventually, things will stop with a ?SYNTAX ERROR or ?OUT OF DATA printed between the two lines, and the tape should stop. (If it doesn't, stop it with the RUN-STOP key). The merge is now complete. Type CLOSE 1 to close the file.

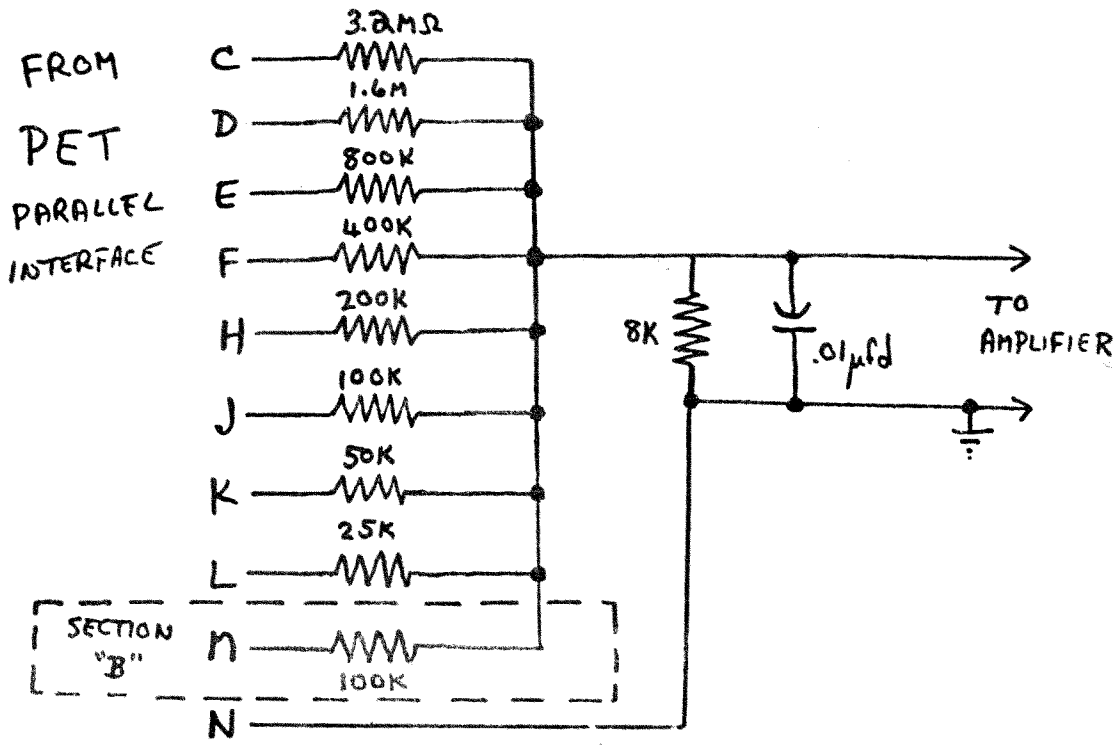
(The ?"POKE611,0" in paragraph 5 may be unnecessary .. I put it there to guard against a processor crash situation I encountered during early testing ...)

Jim has also submitted a schematic of a circuit he demons demonstrated at the first meeting of the Toronto PET Users Group. The schematic has been published in the club newsletter and Lyman Duggan suggested I pass it on to Transactor subscribers. Thanx again Jim. -Karl J.

Poor Man's D/A Converter

Cheap; good for generating Chamberlin/style music. Precision resistors are preferred, but most anything will generate a recognizable sound.

Section B of the diagram supports the CB2 sound effects - so that one interface board covers most sound requirements.



The capacitor provides some reduction of the sampling frequency (when generating music) .. tone controls on the amplifier will also help, if available.

The HUH "Petunia" gives a high quality equivalent of this converter. Toronto price - about $40.

Reference: see BYTE, September 1977, lengthy article by Hal Chamberlin on computer-generated music. 6502 programs are given.

The "ON...GOTO" Statement

e.g. 100 ON I GOTO 10, 20, 146, 2040

The above statement, when encountered, will cause program execution to branch to the line indicated by the I'th number after the GOTO. That is:

```
If I=1    GOTO 10
If I=2    GOTO 20
If I=3    GOTO 146
If I=4.4  GOTO 2040
```

I is truncated to an integer value. If I is equal to zero, or attempts to select a non-existent line (greater than or equal to 5), in this case the statement after the ON...GOTO is executed. As many line numbers as will fit on a line can follow an ON...GOTO. Thus the main purpose is to eliminate successive IF...GOTO statements and save on memory consumption.

e.g. 200 ON SGN(X)+2 GOTO 40, 50, 310

In this case, execution will branch to line 40 if the SGN(X) expression is less than zero, line 50 if equal to zero and line 310 if equal to one.

When using expressions in the ON...GOTO statement, do not allow the final result to be negative. Implement an ABS function into the expression, else an ILLEGAL QUANTITY ERROR will result.

The "ON...GOSUB" Statement

Identical to the ON...GOTO statement except that a subroutine call is executed instead of an absolute GOTO. On return from the subroutine, execution continues at the line following the ON...GOSUB.

A Short Note On Subroutines

Jumping out of a subroutine can be HAZARDOUS! That is, subroutines containing IF...GOTO's or ON...GOTO's can cause an OUT OF MEMORY ERROR for this reason: When a program encounters a GOSUB, the machine loads the return address into a stack area. The subroutine is then executed and let's say 'jumps out' on a GOTO statement within the subroutine. Thus the RETURN statement is never executed and the return address is not unloaded from the stack. The stack will fill to the limit and the error message results.

To avoid this problem the ON...GOTO statement can be used instead of RETURN. Thus all GOSUB statements directed at that particular subroutine must also be replaced with absolute GOTO's. By implementing a control variable just prior to the GOTO, the return addresses can be placed after the ON...GOTO in the subroutine and the hazard is eliminated. For example:

```
100  C=1:GOTO 5000                Subroutine address
110  REM RETURN FROM SUB.

500  C=2:GOTO 5000                Same subroutine
510  REM RETURN FROM SUB.



5000 REM SUBROUTINE
5010 ┐
 .   │
 .   │   'SUBROUTINE'
 .   │
 .   │
5080 ┘
5090 ON C GOTO 110, 510           Instead of RETURN
```

FAILSAFING

Recently I have received a number of inquiries on how to avoid entering the command mode by hitting "RETURN" after an input statement without data entry. I know of three such methods:

1. The "GET" loop instead

e.g.
```
20 GET A$: IF A$=" " THEN 20
30 PRINT A$
```

The above routine will loop continuously in line 20 until a key is depressed. Once entry is made, the routine will print the entry be it alphabetic, numeric or graphic character. Use of the numeric variable is confusing because even if no key has been struck, the value returned is zero. That is:

```
20 GET A: IF STR$(A)=" " THEN 20
30 PRINT A
```

will return 0 immediately.

2. Forced Input

If after an INPUT statement you arrange an invalid input to the right of the '?', hitting "RETURN" will result in a ? REDO FROM START and go back to the INPUT statement. For example:

```
10 INPUT "A VALUE ØØØØ ← clclclcl";A
20 PRINT A
```

The cursor is left 2 character spaces beyond the '?'. Therefore you must arrange your 'invalid input' such that it will be erased by the entry else it will be included (be it to the right or left of the entry), and a ?REDO FROM START will be returned.

The above routine does not work with string variables because ' ▚ ' will be accepted as an alphabetic input if "RETURN" is hit. Therefore a test statement must be added:

```
10 INPUT "A CHAR. ØØØØ ▚ clclclcl";A$
15 IF A$ = " ▚ " THEN 10
20 PRINT A$
```

3. Opening the Keyboard as a File

By assigning the keyboard a file number using the OPEN statement and opening that file for reading, all input statements will accept data only as an entry. The following program will demonstrate this.

```
1 OPEN 1,0,0 (last 0 optional)
.
.
.
.
500 PRINT "A VALUE?";
510 INPUT #1,A$
520 PRINT: PRINT A$
530 END
```

The open statement may be placed at the very beginning of the program and may even be line 0. Other programming can be inserted (between 1 and 500). The preceding words and '?' must be displayed using a PRINT statement (line 500). The double PRINT's in line 520 are required to get A$ to print on the next screen line. Otherwise A$ will be displayed just to the right of the entry.

If anyone should find bugs with this due to not closing the file or otherwise, please let me know so that I may pass it on.

Karl J.